Nova Scotia Information and Communication Technology 5th Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Nova Scotia Information and Communication Technology 5th Grade Course** introduces students to foundational programming concepts through **Scratch**, a block-based programming language. Students will develop computational thinking and problem-solving skills while learning to create interactive projects, animations, and games. This course emphasizes creativity and collaboration, providing students with a solid base in computer science concepts and digital literacy.

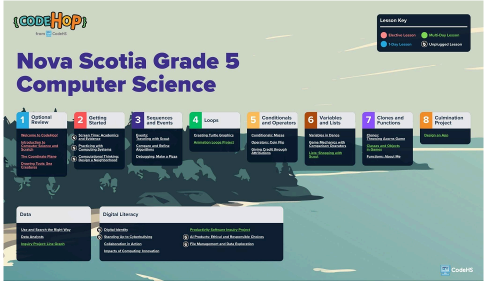
Learning Environment: This course is designed to be teacher-led, with ready-to-use lesson plans that follow a structured format: **Introduction, Guided Practice, Independent Practice, Extension, and Reflection**. Lessons are built with spiral review to reinforce key concepts and culminate in engaging projects to showcase student understanding.

The lessons are delivered in an **"I do, we do, you do"** format, ensuring a gradual release of responsibility and fostering confidence in students as they learn. Teachers can adapt the content to fit their schedule and instructional needs. The concepts taught in this course spiral across grade levels, ensuring that students can revisit and build upon their understanding year after year, even if all lessons are not completed within a single year. The course includes a total of 36 **contact hours**, with each lesson approximately 45 minutes long. This provides a full school year of material if teaching one lesson per week.

Programming Environment: Students will write and run programs in **Scratch** embedded and saved in students' accounts. The environment supports interactive, hands-on programming, enabling students to create and debug projects in a user-friendly interface.

Prerequisites: There are no prerequisites for this course. It is designed to support all learners, regardless of prior computer science experience.

More Information: Browse the content of this course at https://codehs.com/course/NS_5/overview



{CODEHOP} from CodeHS

Nova Scotia Grade 5 Computer Science

Lesson Key
● Elective Lesson
● Multi-Day Lesson
● 1-Day Lesson
Unplugged Lesson

1 Optional Review
- Welcome to CodeHop!
- Introduction to Computer Science and Scratch
- The Coordinate Plane
- Drawing Tools: Sea Creatures

2 Getting Started
- Screen Time: Academics and Evidence
- Practicing with Computing Systems
- Computational Thinking: Design a Neighborhood

3 Sequences and Events
- Events: Traveling with Scout
- Compare and Refine Algorithms
- Debugging: Make a Pizza

4 Loops
- Creating Turtle Graphics
- Animation Loops Project

5 Conditionals and Operators
- Conditionals: Mazes
- Operators: Coin Flip
- Giving Credit through Attributions

6 Variables and Lists
- Variables in Dance
- Game Mechanics with Comparison Operators
- Lists: Shopping with Scout

7 Clones and Functions
- Clones: Throwing Acorns Game
- Classes and Objects in Games
- Functions: About Me

8 Culmination Project
- Design an App

Data
- Use and Search the Right Way
- Data Analysts
- Inquiry Project: Line Graph

Digital Literacy
- Digital Identity
- Standing Up to Cyberbullying
- Collaboration in Action
- Impacts of Computing: Innovation
- Productivity Software Inquiry Project
- AI Products: Ethical and Responsible Choices
- File Management and Data Exploration

CodeHS

A clickable PDF can be found at https://codehs.com/NS-P-5Roadmaps

Course Breakdown

Optional Review

In this unit, students review key computer science concepts by logging into the Playground, creating simple Scratch programs, using the coordinate plane for animation, and customizing projects with Scratch's drawing tools.

Objectives / Topics Covered	● Log in and navigate the Playground. ● Create simple Scratch programs. ● Use the coordinate plane to position sprites.
Lessons	**Welcome to CodeHop!** ● Learn how to log in and navigate the Playground. **Introduction to Computer Science and Scratch** ● Define basic computer science vocabulary and create a simple Scratch program to practice foundational coding skills. **The Coordinate Plane** ● Use the coordinate plane in Scratch to position sprites and create an open-ended animation. **Drawing Tools: Sea Creatures** ● Use image editing tools to create and program deep sea creatures.

Unit 1: Getting Started (3 weeks)

In this unit, students learn to use technology responsibly by creating a personal screen time plan and applying computational thinking to design a neighborhood.

Objectives / Topics Covered	● Understand the importance of balancing screen time and using technology responsibly. ● Apply computational thinking skills. ● Explore how to interact with and operate basic computing systems.
Lessons	**Screen Time: Academics and Evidence** ● Create a healthy screen-time plan and explain how screen time affects learning. **Practicing with Computing Systems** ● Interact with and troubleshoot basic computing systems. **Computational Thinking: Design a Neighborhood** ● Use computational thinking skills—like breaking down tasks and identifying patterns—to design a neighborhood layout.

Unit 2: Sequences and Events (2 weeks)

In this unit, students build foundational programming skills by using events, comparing and refining algorithms for efficiency, and developing debugging strategies to identify and fix problems in a program.

Objectives / Topics Covered	● Use events to control when actions occur in a program. ● Compare and refine multiple algorithms to choose the most effective solution for a task. ● Decompose programs to identify and fix errors through debugging.
Lessons	**Events: Traveling with Scout** ● Use events in a program. **Compare and Refine Algorithms** ● Explore multiple algorithms for completing the same task and refine them to choose the most efficient solution.

Unit 3: Loops (3 weeks)

In this unit, students explore how loops can be used to create visual effects by programming turtle graphics and building multi-scene animations using repeat blocks.

Objectives / Topics Covered	● Use repeat loops to simplify and organize code.
Lessons	**Creating Turtle Graphics** ● Use the pen tool and loops in Scratch to create turtle-style drawings with repeated patterns. **Animation Loops Project (2 part lesson)** ● Use repeat loop blocks to build a multi-scene animation, demonstrating how loops can simplify and organize repeated actions.

Unit 4: Conditionals and Operators (3 weeks)

Students explore conditionals to control the behavior of sprites in their programs. Students will use these skills to add effects to their programs and to develop interactive games.

Objectives / Topics Covered	● Use conditionals to control decision-making in programs. ● Use operators as conditions in a program.
Lessons	**Conditionals: Mazes** ● Create a Scratch program that uses conditionals to guide characters through a maze. **Operators: Coin Flip** ● Use operators to program a coin flipping game. **Giving Credit Through Attributions** ● Learn to give proper credit when using or remixing programs and images to support respectful digital content use.

Unit 5: Variables and Lists (4 weeks)

Students explore variables to store and change values while a program is running. Students will also use lists to store multiple pieces of data.

Objectives / Topics Covered	● Learn how variables store information. ● Create and change the value of variables.
Lessons	**Variables in Dance** ● Use variables to control features like pitch and speed in a dance animation program. **Game Mechanics with Comparison Operators** ● Apply comparison operators and variables to program game-ending conditions and logic. **Lists: Shopping with Scout (2 part lesson)** ● Use variables, lists, and operators to create a shopping simulator.

Unit 6: Clones and Functions (4 weeks)

In this unit, students explore advanced programming concepts by using clones to build interactive games, learning about classes and objects, and applying randomization to create variety in gameplay. They also create and use functions with inputs to organize code and personalize projects, reinforcing the value of reusable code structures.

Objectives / Topics Covered	● Use clones to create repeated sprites. ● Learn about classes and objects by programming games that modify object properties. ● Create and use functions to organize and reuse code efficiently.
Lessons	**Clones: Throwing Acorns Game** ● Create an interactive game using clones to throw acorns and manage repeated sprite actions. **Classes and Objects in Games (2 part lesson)** ● Learn about classes and objects by creating a game that uses randomizers to change object characteristics. **Functions: About Me** ● Create and use a function with input to personalize a program that shares information about the student.

Unit 7: Culmination Project (3 weeks)

In this unit, students apply the design thinking process to create an app that addresses a specific user need, integrating their programming knowledge with problem-solving, creativity, and user-centered design.

Objectives / Topics Covered	● Combine creativity, problem-solving, and programming skills to develop a user-centered digital project.
Lessons	**Design an App (3 part lesson)** ● Use the design thinking process to plan and design an app that addresses a specific user need, combining creativity, problem-solving, and coding skills.

Unit 8: Data (4 weeks)

Students will explore how to search for, analyze, collect, and present data to answer questions and solve problems.

Objectives / Topics Covered	● Search for information responsibly online. ● Evaluate and analyze data. ● Collect data and display results visually.
Lessons	**Use and Search the Right Way** ● Research information to answer questions online and give credit to original sources. **Data Analysts** ● Analyze data to draw conclusions and make predictions. **Inquiry Project: Line Graph (2 part lesson)** ● Follow the inquiry process to collect data and modify a program to display the results.

Unit 9: Digital Literacy (10 weeks)

In this unit, students build digital literacy by practicing responsible online collaboration and learning how to give proper attribution when using digital content. They also investigate data through an inquiry project, study how network protocols support data transfer, and begin to understand the broader social and technical impacts of computing.

Objectives / Topics Covered	● Collaborate effectively online. ● Understand how to spread positivity online. ● Explore current, past, and new technologies.
Lessons	**Digital Responsibility: Owning Your Choices** ● This lesson is coming soon!

	Digital Reputation and Identity ● This lesson is coming soon! **Collaborating Globally (2 part lesson)** ● Collaborate with others digitally to improve a program. **Impacts of Computing: Innovation** ● Create a Scratch project to compare past and present versions of technology. **Productivity Software Inquiry Project (3 part lesson)** ● Use productivity software to organize research and present information to others. **AI Products: Ethical and Responsible Choices** ● Analyze the benefits and challenges of various AI products from multiple perspectives. **File Management and Data Exploration** ● Explore how to save, organize, and manage digital files effectively. ● *Note: the extension must be completed to meet all standards in this lesson.*

Nebraska Technology 5th Grade Course Supplemental Materials

Resources	**Description**
Parent Welcome Letter (Spanish)	Send this letter home to introduce families to their new computer science curriculum.
Warm-Up Activities	This warm-up activity slide deck provides 5-10 minute problems aligned with computer science skills to engage students at the start of class, allowing teachers to preview or review concepts with answer keys and discussion tips included in the Speaker Notes.
Program Self-Assessment (Spanish)	This is a student self-assessment tool designed to help K-6 learners reflect on their programming projects, evaluate their skills in algorithms, debugging, collaboration, and reflection, and set goals for improvement.
Peer Review Resources (Spanish)	This provides structured worksheets to facilitate student feedback during collaborative coding projects. It encourages reflection by guiding students to highlight successes, ask questions, and offer constructive feedback on their partner's work.
Lesson Reflection & Computational Thinking (Spanish)	This guides students in engaging with computational thinking concepts, preparing for discussions, reflecting on lessons, and applying their learning to real-world problem-solving.
Design-Your-Own-Lesson Scratch Templates	Empower your students to explore and express their knowledge creatively with our versatile Scratch graphic organizer templates. Designed with adaptability and ease of use in mind, these interactive tools transform any subject into an engaging, hands-on learning experience.
All of these resources and more are found on the **CodeHop Resources Page**.	